(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 July 2001 (19.07.2001)

PCT

(10) International Publication Number
WO 01/52060 A1

(51) International Patent Classification[7]: G06F 9/38, 9/30

(21) International Application Number: PCT/EP00/00259

(22) International Filing Date: 14 January 2000 (14.01.2000)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant and
(72) Inventor: THEIS, Jean-Paul [LU/LU]; 1, Porte des Ardennes, L-9145 Erpeldange (LU).

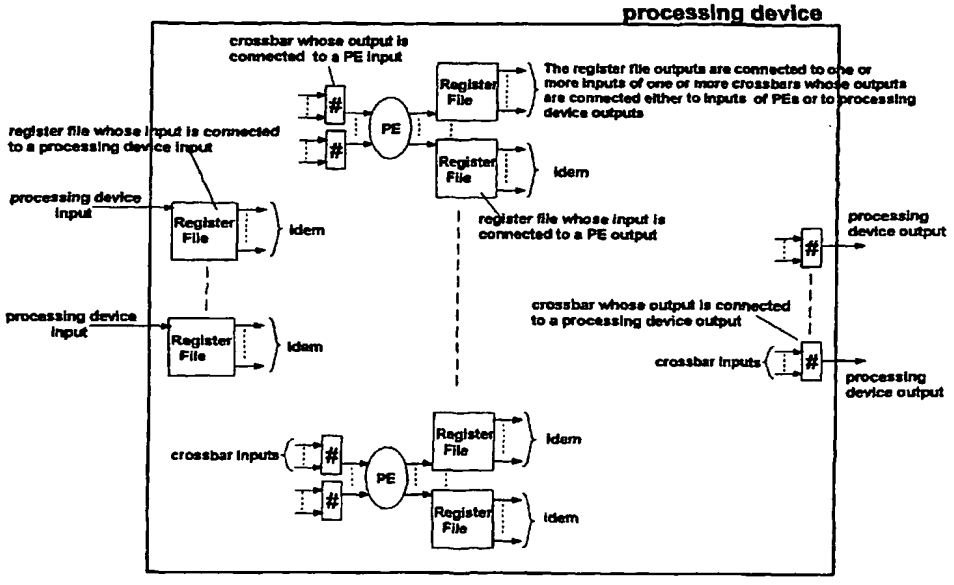
(81) Designated States *(national)*: JP, US.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A DATA PROCESSING DEVICE WITH DISTRIBUTED REGISTER FILE



(57) Abstract: The present invention introduces data path architectures of processing devices with a distributed register file. These data path architectures are obtained by applying a set of building rules to a set of building blocks. The number of register files corresponds to the number of processing device inputs and processing element outputs. Specific rules for connecting the register files to the processing elements via distributed crossbars are given. Arrays of processing devices are considered as well. Data path architectures of processing devices used in these arrays slightly differ from those of stand alone processing devices.

WO 01/52060 A1

BEST AVAILABLE COPY

A data processing device with distributed register file

1. Field of the invention

The present invention relates to the field of architecture design of data processing devices. More specifically, the invention is dealing with architecture design issues at register-transfer level and is focusing on data path architectures of processing devices.

2. Conventions, definition of terms, terminology

First, it should be noted that in the literature the two expressions 'distributed register file' and 'distributed register files' (files with an 's') are used synonymously and stand for two or more register files.

The term 'data processing device' has a very broad meaning and can stand for terms like (micro)processor, micro-controller, central processing unit (CPU), digital signal processor (DSP), application specific integrated circuit (ASIC), application specific standard product (ASSP), application specific instruction set processor (ASIP). As mentioned before, the present invention is dealing with architecture design issues at register-transfer-level. A register-transfer level architecture of a processing device can be thought of as consisting of a limited number of elementary building blocks with which the processing device is built up. The register transfer-level architecture of a processing device typically consists of Processing Elements (PEs), register files, busses, crossbars and a control unit which are arranged and connected to each other in a well defined manner. The way how these building blocks are arranged and connected together determines the features of an architecture of such a processing device. The term 'PE' is frequently used in the same sense as 'data procesing device'. However in the text that follows, the term 'PE' has a more restricted meaning and will represent, unless specified otherwise, either Arithmetic Logic Units (ALUs), floating point units (FPUs) or other functional units (FUs) of a processing device. A crossbar is a building block that makes connections between its inputs and outputs. A fully connected crossbar is able to connect any input to one, more or even all outputs. A partially connected crossbar is able to connect any input to one or more but not all outputs. Multiplexers/demultiplexers are crossbars with one input/output and one or more outputs/inputs respectively. The meaning of the other before mentioned building blocks is identical to the one normally described in the literature.

The (register-transfer-level) data path architecture of a processing device comprises only building blocks directly involved in the data processing, e.g. PEs, register files, busses and crossbars, but not any

control units used to control the building blocks of the data path. Therefore in all the following figures, control signals for crossbars, PEs and register files will only be shown when they are relevant in the context of the present invention. Furthermore, in all the figures that follow, arrows represent either bussed connections between building blocks or bussed inputs and bussed outputs of building blocks and processing devices, where the bus width of a bussed connection or of a bussed input/output is equal to one or more bits. Unless specified otherwise, all the inputs and outputs of building blocks and of a processing device itself refer to data and not to control signals. It is assumed that all the control signals for all the building blocks are generated from one or more control units of the processing device, these control units typically comprising instruction decode and execution units as well as memory management units. Control signals for register files f.ex. determine the addresses of the register locations to/from which data are written/read respectively or represent clocking signals. Register file inputs are also called write ports and register file outputs are also called read ports. Read/write ports may have simultaneous access to all register locations in the register file. Control signals for PEs f.ex. select the operations to be performed. Control signals for crossbars determine the connections to be made between crossbar inputs and crossbar outputs.

3. Prior Art

Before investigating the prior art in processor architectures with a distributed register file, it is worthwhile to have in mind the data path architecture of a 'conventional' processor with a single register file as it is used in today's microprocessors and as shown in figure 1. It is characterized by the fact that all the PE outputs are connected to the same one register file and that all the PEs may have simultaneous access (for reading and writing data) to any register location in the register file.

Figures 2, 3 and 4 can be used to retrace briefly the evolutionary steps of register-transfer-level architectures of processing devices with a distributed register file.

Figure 2 shows one of the first data path architectures of a processing device with a distributed register file. It was called the polycyclic processor and was developed by ESL Inc. in the early 80's. The data path architecture at register-transfer-level is shown and consists of a set of PEs whose inputs and outputs are connected to a crossbar with delay elements at each cross point. The delay elements can be thought of as a particular implementation of a register file. PEs, crossbar and delay elements are connected in the following way : (1) each PE (data) output is connected to as many independent cross points in a row of the crossbar as there are PE inputs (2) each PE input is connected to and selected out of as many cross points in a column as there are PE outputs. For the configuration as shown in figure 2 with two PEs having each 2 inputs and 1 output, this implies a fully connected crossbar with 4X2 independent cross points, in other words 2 rows with each 4 cross points or equivalently 4 columns with each 2 cross points, and with as many delay elements as cross points. Note that this is drawn symbolically in figure 2. The detailed architecture of the crossbar with the delay elements is not shown.

It is important to note that ESL Inc. had microprocessors in mind when speaking of PEs. Therefore, the crossbar with delay elements is first of all an efficient method of exchanging data between microprocessors, hence an efficient method to build multi-processor systems. A next step in the evolution of data path architectures with a distributed register file consisted in integrating the crossbar with delay elements as shown in figure 2 directly into the data path architecture of a microprocessor. This step was done in the data path architecture (again at register-transfer-level) of a video signal processor which was developed in the late 80's by Philips Research and which is shown in figure 3.

Although the terminology used in figure 3 is slightly different from that used in figure 2, the building blocks in question and the way in which they are connected together are identical : in figure 3 the crossbar is called a switch matrix, the delay elements are called Silos, the PEs are called ALEs (Arithmetic Logic Elements), where ALE is yet another word for ALU. In figure 3, the Silos are used for slightly different data storage purposes : 1) as Memory Elements (MEs) which contain in addition to the Silos conventional memory for program data and logic for address calculation 2) as Buffer Elements (BEs) for buffering data 3) as Output Elements (OEs) for buffering data before they leave the processor. As mentioned above, the way in which these building blocks are connected together is the same as in figure 2, with the only difference lying in a more explicit separation and drawing of crossbar (switch matrix) and delay elements (Silos).

Finally, replacing delay elements (Silos) with conventional register files leads to a data path architecture with distributed register file as shown in figure 4. In figure 4, the PEs can be of different type, and with several data inputs and data outputs. The register files can be of different type as well as, like f.ex. stacks, FIFOs and register files with rotating property where the data rotate in the register file, and they may have several read and write ports from and to which data can be read and written simultaneously. The crossbar can be fully or only partially connected. Furthermore, outputs of register files may be connected to PE inputs and/or to processor outputs.

It is interesting to see that the way in which the building blocks, consisting of crossbar, register files (Silos, delay elements) and PEs (microprocessors, ALUs, ALEs), are connected together in figures 2, 3 and 4 appears to be identical and is based on the following rules : 1) take the data outputs of the PEs (ALUs, ALEs) and connect them to the crossbar inputs 2) take the crossbar outputs and connect them to the inputs of the register file, delay elements and Silos 3) take the outputs of the register files, delay elements and Silos and connect them to the (data) inputs of the PEs.

Before closing this section over the prior art in data path architectures of processing devices with a distributed register file, their short comings and major points for improvement will be shortly discussed.

Two major shortcomings of a 'conventional' data path architecture with a single register file are the VLSI design challenge of the single register file and the power consumption of the single register file. Today's

microprocessors (f.ex. Pentium, PowerPC) have single register files containing at least 128 80-bit floating point registers and having at least 4 read and 4 write ports. This leads to a big silicon area of the register file which leads in its turn to an increase in read/write/access cycle times due to long wire lines to be charged and discharged. In order to compensate for this effect, special design techniques have to be utilized in order to keep the read/write/access cycle times down to an acceptable level. This however, together with the big silicon area, goes to the detriment of power consumption and therefore big single register files with multiple read/write ports are not very power efficient.

Data path architectures with distributed register file try to overcome these shortcomings by using several and smaller register files with only a few read/write ports. All these register files together are of about the same size as a big single register file. However in case of a data path architecture like in figure 4, the prize that is paid to overcome the problems linked to a single register file consists in bigger code size. This is due to the fact that data path architectures with distributed register files are typically VLIW processor architectures where a compiler is optimizing the program code statically in order to optimally exploit the multiple register. For a certain number of reasons however, the program code of VLIW processors is typically twice as large as for 'conventional ' processors (processing devices) with a single register file.

Another major point for improvement of processing devices with a single register file as well as with a distributed register file concerns the implementation costs. It was already mentioned that for a certain number of reasons single register files are always of big size and therefore have high implementation costs in term of silicon area. However the same is true for distributed register files if they are used as in figure 4 because they require a large crossbar to make the connections between the multiple register files and the PEs.

It is the goal of the present invention to overcome these shortcomings of existing data path architectures with a single register file as well as with a distributed register file.

4. Brief description of the drawings

Figure 1 shows the data path architecture of a 'conventional' processor with a single register file.
Figure 2 shows the data path architecture of the polycyclic processor developed by ESL Inc.
Figure 3 shows the data path architecture of a video signal processor developed by Philips Research.
Figure 4 shows the data path architecture of a processor with a distributed register file according to the prior art.
Figure 5 shows the data path architecture of a processing device with a distributed register file based on the present invention.
Figure 6 shows a specific example of the data path architecture of a processing device with a distributed register file based on the present invention.

Figure 7 shows two variants of a specific type of register file containing a shift register connected to a crossbar. One variant of this type of register file is shown at lower right, the other variant is shown at lower left.
Figure 8 shows a specific example of an array of processing devices built up according to the rules based on the present invention.
Figure 9 shows two processing devices of an array and visualizes the rules concerning a) processing device inputs which are connected to an array input and b) processing device inputs which are connected to an output of a processing device of the array.

5. Detailed description of the drawings

The main aspects of the present invention are described by referring to the figures mentioned in this section.

Data path architectures of processing devices with a distributed register file based on the present invention differ significantly from the data path architectures of the prior art and are obtained by applying a set of building rules to a set of building blocks. The differences with the prior art will become clear when discussing these building rules.

Considered is a processing device comprising one or more inputs, one or more outputs and one or more processing elements, each processing element having one or more inputs and one or more outputs. In the following, unless mentioned explicitly, the terms 'data path architecture', 'crossbar', 'register file' and 'processing element' always refer to the considered processing device.

The first type of data path architecture of a processing device based on the present invention contains :

(a) as many register files as there are processing device inputs and processing element outputs, where processing element outputs correspond to all the outputs of all the processing elements of the considered processing device and where all the register files have each one input and one or more outputs

(b) as many crossbars as there are processing device outputs and processing element inputs, where processing element inputs correspond to all the inputs of all the processing elements of the considered processing device and where all the crossbars have each one output and one or more inputs

and has a register-transfer-level data path architecture which is built up according to the following rules :

(c) each processing device input is connected to the input of a register file

(d) any processing device input and any other processing device input are not connected to the same input of a register file

(e) each output of each processing element is connected to the input of a register file

(f) any output of any processing element and any other output of any processing element are not connected to the same input of a register file

(g) the input of each register file is connected either to an output of a processing element or to a processing device input

(h) the input of any register file and the input of any other register file are not connected to a same output of any processing element

(i) the input of any register file and the input of any other register file are not connected to a same processing device input

(j) each output of each register file is connected to an input of a crossbar

(k) any output of any register file and any other output of any other register file are not connected to a same input of a crossbar

(l) each input of each crossbar is connected to an output of a register file

(m) any input of any crossbar and any input of any other crossbar are not connected to the same output of any register file

(n) each processing device output is connected to the output of a crossbar

(o) each input of each processing element is connected to the output of a crossbar

(p) any processing device output and any other processing device output are not connected to the same output of a crossbar

(q) any processing element input and any other processing element input are not connected to the same output of a crossbar

(r) the output of each crossbar is connected either to a processing device output or to an input of a processing element

(s) the output of any crossbar and the output of any other crossbar are neither connected to a same processing device output nor to a same input of a processing element

Note that the rules as described in (c)-(s) do not imply that each output of each register file has necessarily to be connected to all the inputs of all the crossbars. It is left up to the designer to decide which connections between outputs of register files and inputs of crossbars he wants to implement. Therefore, any crossbar has as many inputs as there are outputs of register files connected to that crossbar.

Furthermore, it should be noted that normally all inputs and all outputs of all register files, crossbars and processing elements as well as all processing device inputs and all processing device outputs have the same bus width, the bus width being equal to one or more bits; in other words, all connections as specified in (c)-(s) have the same bus width, the bus width being equal to one or more bits. However it is also conceivable that the bus width differs from connection to connection and from input/output to input/output of building blocks. In case of PEs, the bus width of the PE inputs may well be different from the bus width of the PE outputs, depending of the operations that are performed in the PEs. In case of a crossbar, the connections may be done according to some rule, f.ex. connecting only the most/least significant bits of an input whose bus width is wider than the one of a crossbar output to which the

connection is done or f.ex. filling the most/least significant bits of a crossbar output, whose bus width is wider than the one of a crossbar input to which the connection is done, with some specific values. In case of a register file, the data values appearing on one or more inputs of the register file may be written/read into/from register locations according to similar rules as for the connections to be done inside a crossbar, depending on the bus width of the register file inputs, of the register cells contained in the register file and of the register file outputs.

A processing device with a data path architecture built up according to the rules mentioned above is shown in figure 5. Figure 5 aims at visualizing the above rules, therefore the number of processing device inputs and outputs, the number of PEs as well as the number of PE inputs and PE outputs is not further specified. In contrast, figure 6 shows a specific example of a processing device with such a data path architecture : it contains two PEs, two processing device inputs and two processing device outputs. Each PE has two inputs and two outputs. Register files have either one, two or three outputs. Furthermore the number of existing connections between outputs of register file and inputs of crossbars differ from register file to register file and from crossbar to crossbar, in other words not all connections that are allowed by the rules are effectively realized.

The second type of data path architecture of a processing device based on the present invention slightly differs from the first type in the way that this second type of data path architecture contains one or more register files of a same type, this type of register file being shown in figure 7 and denoted by ' SR + # '. There are basically two slightly different variants of this type of register file, one variant shown at lower right in figure 7 and the other variant shown at lower left in figure 7. Both variants contain a shift register and a crossbar, the crossbar being denoted by '#' in figure 7, and where

(a) the shift register contains one or more register cells
(b) the shift register has one input and as many outputs as there are register cells contained in the shift register, each register cell having one input and one output
(c) the crossbar is either partially or fully connected and has as many outputs as there are register file outputs
(d) in case of one variant, the crossbar has as many inputs as there are register cells contained the shift register
(e) in case of the other variant, the crossbar has as many inputs as the number obtained by incrementing by one the number of register cells contained in the shift register
(f) in case of one variant, the register file input is connected to the input of the shift register
(g) in case of the other variant, the register file input is connected to the input of the shift register and to an input of the crossbar
(h) the input of the shift register is connected to the input of the first register cell of the shift register
(i) the inputs and outputs of the register cells of the shift register are connected in such a way as to form a shift register

(j) the output of each register cell of the shift register is connected to a shift register output
(k) the output of any register cell of the shift register and the output of any other register cell of the shift register are not connected to a same shift register output
(l) each shift register output is connected to an input of the crossbar
(m) any shift register output and any other shift register output are not connected to a same input of the crossbar
(n) in case of one variant, each input of the crossbar is connected to a shift register output
(o) in case of the other variant, each input of the crossbar is connected either to a shift register output or to the register file input
(p) any input of the crossbar and any other input of the crossbar are neither connected to the same shift register output nor to the register file input
(q) each output of the crossbar is connected to a register file output
(r) any output of the crossbar and any other output of the crossbar are not connected to a same register file output
(s) each register file output is connected to a crossbar output
(t) any register file output and any other register file output are not connected to a same crossbar output

The difference between the two variants lies in the fact that in case of the variant shown at lower left in figure 7, the register file input can directly be forwarded to one or more register file outputs without traversing a cell of the shift register. Furthermore, the shift register contained in the register file may have a gated clock input, in other words the contents of the register cells are only then shifted by one position in the shift direction within every clock cycle of some clock used in the processing device if some signals generated in the control unit(s) of the processing device have a specific value. The value of these signals may change from clock cycle to clock cycle of some clock used in the processing device and generally depend on the program code, on the instructions that are executed by the processing device, on results of operations performed by the PEs and on data values stored in the register files. As mentioned before, a shift register with m cells has a 'shift direction', in other words there exists a increasing order of register cells labeled 1,2 ...m such that when the shift register is clocked the content of register cell with label i is shifted into register cell with label i+1, for i=1,2,....m-1. The first cell of the shift register is the register cell with label 1, the last cell of the shift register is the cell with label m. Note that concerning the bus width of any connections between any inputs and outputs of the shift register, of the crossbar, of any register cell of the shift register and of the register file itself the same remark holds as for the connections done inside a processing device with a data path architecture of the first type as described above.

The present invention is also dealing with arrays of processing devices. The data path architecture of the processing devices used in these arrays is closely related to the data path architecture of the first and second type as described above.

Considered is an array comprising two or more processing devices and one or more array inputs and one or more array outputs. Each processing device of the considered array has one or more inputs and one or more outputs. In the following, unless mentioned explicitly, the term 'processing device' always refers to the considered array.

The array is built up according to the following rules :

(a) each array input is connected to one or more inputs of one or more processing devices
(b) any array input and any other array input are not connected to a same input of a processing device
(c) each output of each processing device is connected to one or more inputs of one or more processing devices or to one or more array outputs
(d) any output of any processing device and any output of any other processing device are neither connected to a same input of a processing device nor to a same array output

Here again, concerning the bus width of any connections between any inputs and outputs of the array itself and/or of any processing devices of the array, the same remark holds as for the connections done inside a processing device with a data path architecture of the first or second type as described above. Finally, it should be noted that the rules as described in (a)-(d) allow for regular and irregular connections as it is exemplified by the array shown in figure 8.

Furthermore, the rules as described in (a)-(d) do not imply that all possible connections, which are allowed by the rules, between inputs/outputs of processing devices and array inputs/outputs are effectively realized. It is left up to the designer to decide which connections he wants to implement.

The first and second type of data path architecture as described above are used inside 'stand alone' processing devices, in other words processing devices which are not part of an array of several processing devices. The type of data path architecture of processing devices which are part of an array slightly differs from the first and second type of data path architecture of a 'stand alone' processing device. The difference consists in the number of register files used inside each processing device of the array as well as in the way that inputs of register files are connected to processing device inputs. In a few words, the difference is as follows : if an input of any processing device of the considered array is not connected to an output of a processing device of the considered array but is connected to an array input, then it is connected to the input of a register file in the same way as for the data path architecture of the first and second type described above; if an input of any processing device of the considered array is connected to an output of a processing device of the considered array but is not connected to an array input, then it is directly connected to one or more inputs of one or more crossbars of the considered processing device. This rule is visualized in figure 9. Figure 9 shows thereby two processing devices of an array. As one can see, the input of the processing device at the right side, which is connected to an output of the processing device at the left side, is not connected to an input of a register file but directly connected to one or more inputs of one or more crossbars of that processing

device. On the other hand, the input of the processing device at the right side, which is connected to an array input, is connected to an input of a register file in the same way as for the data path architecture of the first or second type described above.

In the following, unless mentioned explicitly, the terms 'processing device input', 'processing device output', crossbar(s)', 'register file(s)' and 'processing element(s)' always refer to the considered processing device.

In detail, this means that each processing device of the considered array contains :

(a) one or more processing device inputs and one or more processing device outputs

(b) one or more processing elements, each processing element having one or more inputs and one or more outputs

(c) as many register files as the considered processing device has processing element outputs and marked processing device inputs, where

1. processing element outputs correspond to all the outputs of all the processing elements of the considered processing device
2. marked processing device inputs correspond to all those inputs of the considered processing device which are connected to an array input
3. each register file has one input and one or more outputs

(d) as many crossbars as there are processing device outputs and processing element inputs, where processing element inputs correspond to the inputs of all the processing elements of the considered processing device and where each crossbar has one output and one or more inputs

and has a register-transfer-level data path architecture which is built up according to the following rules :

(e) each processing device input which is connected to an array input is connected to the input of a register file

(f) each processing device input which is connected to an output of a processing device of the considered array is connected to one or more inputs of one or more crossbars

(g) any processing device input and any other processing device input are neither connected to the same input of a register file nor to a same input of a crossbar.

(h) each output of each processing element is connected to the input of a register file

(i) any output of any processing element and any other output of any processing element are not connected to the same input of a register file

(j) the input of each register file is connected either to an output of a processing element or to a processing device input

(k) the input of any register file and the input of any other register file are not connected to a same output of any processing element

(l) the input of any register file and the input of any other register file are not connected to a same processing device input

(m) each output of each register file is connected to an input of a crossbar
(n) any output of any register file and any other output of any other register file are not connected to a same input of a crossbar
(o) each input of each crossbar is connected either to an output of a register file or to a processing device input
(p) any input of any crossbar and any input of any other crossbar are neither connected to the same output of any register file nor to a same processing device input
(q) each processing device output is connected to the output of a crossbar
(r) each input of each processing element is connected to the output of a crossbar
(s) any processing device output and any other processing device output are not connected to the same output of a crossbar
(t) any processing element input and any other processing element input are not connected to the same output of a crossbar
(u) the output of each crossbar is connected either to a processing device output or to an input of a processing element
(v) the output of any crossbar and the output of any other crossbar are neither connected to a same processing device output nor to a same input of a processing element

Here again, concerning the bus width of any connections between any inputs and outputs of building blocks of a processing device of the array, the same remark holds as for the connections done inside a processing device with a data path architecture of the first or second type described above.

Furthermore, concerning any processing device of the array, the rules as described in (e)-(v) above do not imply that each processing device input (which is connected to an array input) or each output of each register file has necessarily to be connected to all the inputs of all the crossbars. It is left up to the designer to decide which connections between outputs of register files and inputs of crossbars he wants to implement. Therefore, any crossbar has as many inputs as there are outputs of register files and processing device inputs connected to that crossbar.

It should be mentioned that current semiconductor process technology allows to integrate arrays containing several processing devices onto a single chip. The application domain of 'stand alone' processing devices with a data path architecture based on the present invention is the same as the application domain of arrays of processing devices with data path architectures based on the present invention and consists of applications within image/multimedia/signal processing, graphics processing and linear algebra.

Before closing this section, it is important to mention that for a certain number of reasons concerning code density, compiler optimization, power consumption and computing power performance it is particularly interesting to let all the processing elements of a processing device with a data path architecture based on the present invention be of the same type (in other words to let all the processing

elements be of the same type), whether the considered processing device is a 'stand alone' processing device or whether the considered processing device is part of an array of processing devices based on the present invention.

6. Summary of the invention

The present invention concerns a processing device according to claim 1 and an array of processing devices according to claim 8.

Claims

What is claimed is :

1. A processing device comprising :
   (a) one or more processing device inputs
   (b) one or more processing device outputs
   (c) one or more processing elements, each processing element having one or more inputs and one or more outputs
   (d) as many register files as there are processing device inputs and processing element outputs, where processing element outputs correspond to all the outputs of all the processing elements of the considered processing device and where all the register files have each one input and one or more outputs
   (e) as many crossbars as there are processing device outputs and processing element inputs, where processing element inputs correspond to all the inputs of all the processing elements of the considered processing device and where all the crossbars have each one output and one or more inputs

   and having a register-transfer-level data path architecture which is built up according to the following rules :
   (f) each processing device input is connected to the input of a register file
   (g) any processing device input and any other processing device input are not connected to the same input of a register file
   (h) each output of each processing element is connected to the input of a register file
   (i) any output of any processing element and any other output of any processing element are not connected to the same input of a register file
   (j) the input of each register file is connected either to an output of a processing element or to a processing device input
   (k) the input of any register file and the input of any other register file are not connected to a same output of any processing element
   (l) the input of any register file and the input of any other register file are not connected to a same processing device input
   (m) each output of each register file is connected to an input of a crossbar
   (n) any output of any register file and any other output of any other register file are not connected to a same input of a crossbar
   (o) each input of each crossbar is connected to an output of a register file
   (p) any input of any crossbar and any input of any other crossbar are not connected to the same output of any register file
   (q) each processing device output is connected to the output of a crossbar
   (r) each input of each processing element is connected to the output of a crossbar

(s) any processing device output and any other processing device output are not connected to the same output of a crossbar

(t) any processing element input and any other processing element input are not connected to the same output of a crossbar

(u) the output of each crossbar is connected either to a processing device output or to an input of a processing element

(v) the output of any crossbar and the output of any other crossbar are neither connected to a same processing device output nor to a same input of a processing element

2. A processing device as claimed in claim 1, where one or more or all register files are of a same type, this type of register file comprising :

(a) a shift register containing one or more register cells, the shift register having one input and as many outputs as there are register cells contained in the shift register, each register cell having one input and one output

(b) a crossbar which is either partially or fully connected and which has as many inputs as there are register cells contained in the shift register and which has as many outputs as there are register file outputs

and where

(c) the register file input is connected to the input of the shift register

(d) the input of the shift register is connected to the input of the first register cell of the shift register

(e) the inputs and outputs of the register cells of the shift register are connected in such a way as to form a shift register

(f) the output of each register cell of the shift register is connected to a shift register output

(g) the output of any register cell of the shift register and the output of any other register cell of the shift register are not connected to a same shift register output

(h) each shift register output is connected to an input of the crossbar

(i) any shift register output and any other shift register output are not connected to a same input of the crossbar

(j) each input of the crossbar is connected to a shift register output

(k) any input of the crossbar and any other input of the crossbar are neither connected to the same shift register output nor to the register file input

(l) each output of the crossbar is connected to a register file output

(m) any output of the crossbar and any other output of the crossbar are not connected to a same register file output

(n) each register file output is connected to a crossbar output

(o) any register file output and any other register file output are not connected to a same crossbar output

3. A processing device as claimed in claim 1, where one or more or all register files are of a same type, this type of register file comprising :
   (a) a shift register containing one or more register cells, the shift register having one input and as many outputs as there are register cells contained in the shift register, each register cell having one input and one output
   (b) a crossbar which is either partially or fully connected and which has as many inputs as the number obtained by incrementing by one the number of register cells contained in the shift register and as many outputs as there are register file outputs

   and where
   (c) the register file input is connected to the input of the shift register and to an input of the crossbar
   (d) the input of the shift register is connected to the input of the first register cell of the shift register
   (e) the inputs and outputs of the register cells of the shift register are connected in such a way as to form a shift register
   (f) the output of each register cell of the shift register is connected to a shift register output
   (g) the output of any register cell of the shift register and the output of any other register cell of the shift register are not connected to a same shift register output
   (h) each shift register output is connected to an input of the crossbar
   (i) any shift register output and any other shift register output are not connected to a same input of the crossbar
   (j) each input of the crossbar is connected either to a shift register output or to the register file input
   (k) any input of the crossbar and any other input of the crossbar are neither connected to the same shift register output nor to the register file input
   (l) each output of the crossbar is connected to a register file output
   (m) any output of the crossbar and any other output of the crossbar are not connected to a same register file output
   (n) each register file output is connected to a crossbar output
   (o) any register file output and any other register file output are not connected to a same crossbar output

4. A processing device as claimed in claim 2, where the shift register of said type of register file contains at least 4 register cells

5. A processing device as claimed in claim 3, where the shift register of said type of register file contains at least 4 register cells

6. A processing device as claimed in claim 4, where said type of register file has at least two outputs

7. A processing device as claimed in claim 5, where said type of register file has at least two outputs

8. An array comprising :

(a) two or more processing devices, each processing device of the considered array having one or more inputs and one or more outputs

(b) one or more array inputs and one or more array outputs.

where in the following, unless mentioned explicitly, the term 'processing device(s)' always refer to the considered array and where the array is built up according to the following rules :

(c) each array input is connected to one or more inputs of one or more processing devices

(d) any array input and any other array input are not connected to a same input of a processing device

(e) each output of each processing device is connected to one or more inputs of one or more processing devices or to one or more array outputs

(f) any output of any processing device and any output of any other processing device are neither connected to a same input of a processing device nor to a same array output

where in the following, unless mentioned explicitly, the terms 'processing device input', 'processing device output', crossbar(s)', 'register file(s)' and 'processing element(s)' always refer to the considered processing device and where each processing device of the array contains :

(g) one or more processing device inputs and one or more processing device outputs

(h) one or more processing elements, each processing element having one or more inputs and one or more outputs

(i) as many register files as the considered processing device has processing element outputs and marked processing device inputs, where

i. processing element outputs correspond to all the outputs of all the processing elements of the considered processing device

ii. marked processing device inputs correspond to all those inputs of the considered processing device which are connected to an array input

iii. each register file has one input and one or more outputs

(j) as many crossbars as there are processing device outputs and processing element inputs, where processing element inputs correspond to the inputs of all the processing elements of the considered processing device and where each crossbar has one output and one or more inputs

and where each processing device of the array has a register-transfer-level data path architecture which is built up according to the following rules :

(k) each processing device input which is connected to an array input is connected to the input of a register file

(l) each processing device input which is connected to an output of a processing device of the considered array is connected to one or more inputs of one or more crossbars

(m) any processing device input and any other processing device input are neither connected to the same input of a register file nor to a same input of a crossbar

(n) each output of each processing element is connected to the input of a register file

(o) any output of any processing element and any other output of any processing element are not connected to the same input of a register file

(p) the input of each register file is connected either to an output of a processing element or to a processing device input

(q) the input of any register file and the input of any other register file are not connected to a same output of any processing element

(r) the input of any register file and the input of any other register file are not connected to a same processing device input

(s) each output of each register file is connected to an input of a crossbar

(t) any output of any register file and any other output of any other register file are not connected to a same input of a crossbar

(u) each input of each crossbar is connected either to an output of a register file or to a processing device input

(v) any input of any crossbar and any input of any other crossbar are neither connected to the same output of any register file nor to a same processing device input

(w) each processing device output is connected to the output of a crossbar

(x) each input of each processing element is connected to the output of a crossbar

(y) any processing device output and any other processing device output are not connected to the same output of a crossbar

(z) any processing element input and any other processing element input are not connected to the same output of a crossbar

(aa) the output of each crossbar is connected either to a processing device output or to an input of a processing element

(bb) the output of any crossbar and the output of any other crossbar are neither connected to a same processing device output nor to a same input of a processing element

9. An array as claimed in claim 8, where one or more or all register files of all the processing devices of the array are of a same type, this type of register file comprising :

(a) a shift register containing one or more register cells, the shift register having one input and as many outputs as there are register cells contained in the shift register, each register cell having one input and one output

(b) a crossbar which is either partially or fully connected and which has as many inputs as there are register cells contained in the shift register and which has as many outputs as there are register file outputs

and where

(c) the register file input is connected to the input of the shift register

(d) the input of the shift register is connected to the input of the first register cell of the shift register

(e) the inputs and outputs of the register cells of the shift register are connected in such a way as to form a shift register
(f) the output of each register cell of the shift register is connected to a shift register output
(g) the output of any register cell of the shift register and the output of any other register cell of the shift register are not connected to a same shift register output
(h) each shift register output is connected to an input of the crossbar
(i) any shift register output and any other shift register output are not connected to a same output of the crossbar
(j) each input of the crossbar is connected to a shift register output
(k) any input of the crossbar and any other input of the crossbar are neither connected to the same shift register output nor to the register file input
(l) each output of the crossbar is connected to a register file output
(m) any output of the crossbar and any other output of the crossbar are not connected to a same register file output
(n) each register file output is connected to a crossbar output
(o) any register file output and any other register file output are not connected to a same crossbar output

10. An array as claimed in claim 8, where one or more or all register files of all the processing devices of the array are of a same type, this type of register file comprising :
(a) a shift register containing one or more register cells, the shift register having one input and as many outputs as there are register cells contained in the shift register, each register cell having one input and one output
(b) a crossbar which is either partially or fully connected and which has as many inputs as the number obtained by incrementing by one the number of register cells contained in the shift register and as many outputs as there are register file outputs

and where

(c) the register file input is connected to the input of the shift register and to an input of the crossbar
(d) the input of the shift register is connected to the input of the first register cell of the shift register
(e) the inputs and outputs of the register cells of the shift register are connected in such a way as to form a shift register
(f) the output of each register cell of the shift register is connected to a shift register output
(g) the output of any register cell of the shift register and the output of any other register cell of the shift register are not connected to a same shift register output
(h) each shift register output is connected to an input of the crossbar
(i) any shift register output and any other shift register output are not connected to a same input of the crossbar

(j) each input of the crossbar is connected either to a shift register output or to the register file input

(k) any input of the crossbar and any other input of the crossbar are neither connected to the same shift register output nor to the register file input

(l) each output of the crossbar is connected to a register file output

(m) any output of the crossbar and any other output of the crossbar are not connected to a same register file output

(n) each register file output is connected to a crossbar output

(o) any register file output and any other register file output are not connected to a same crossbar output

11. An array as claimed in claim 9, where the shift register of said type of register file contains at least 4 register cells

12. An array as claimed in claim 10, where the shift register of said type of register file contains at least 4 register cells

13. An array as claimed in claim 11, where said type of register file has at least two outputs

14. An array as claimed in claim 12, where said type of register file has at least two outputs

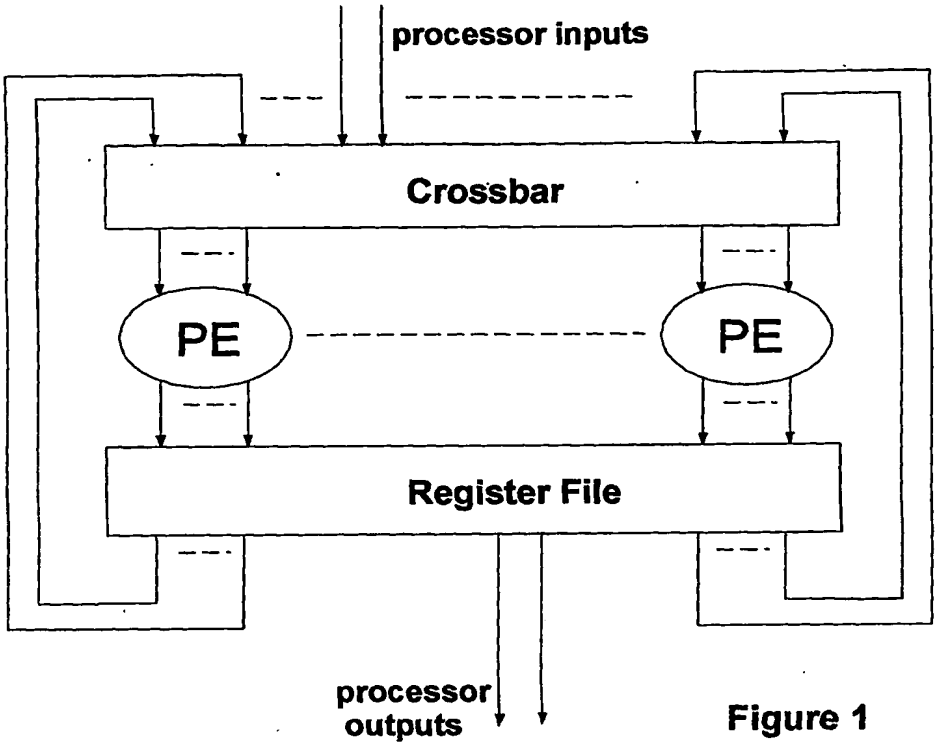


Figure 1

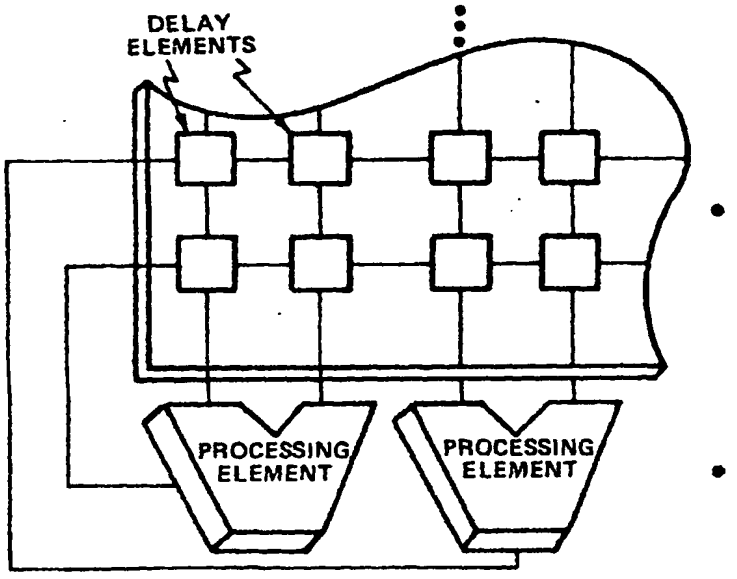


Figure 2

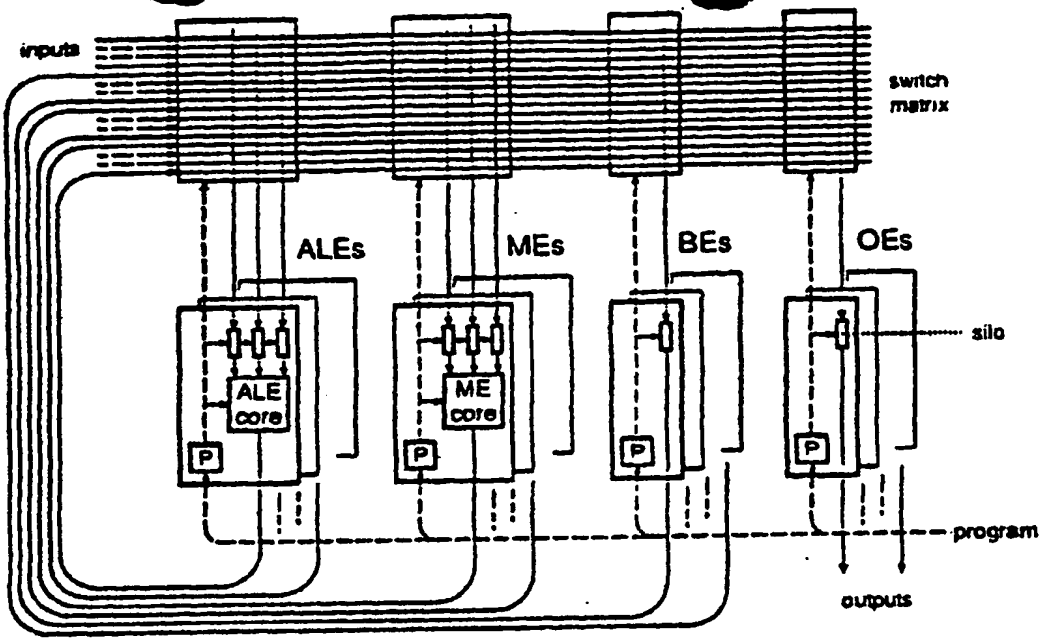


Figure 3

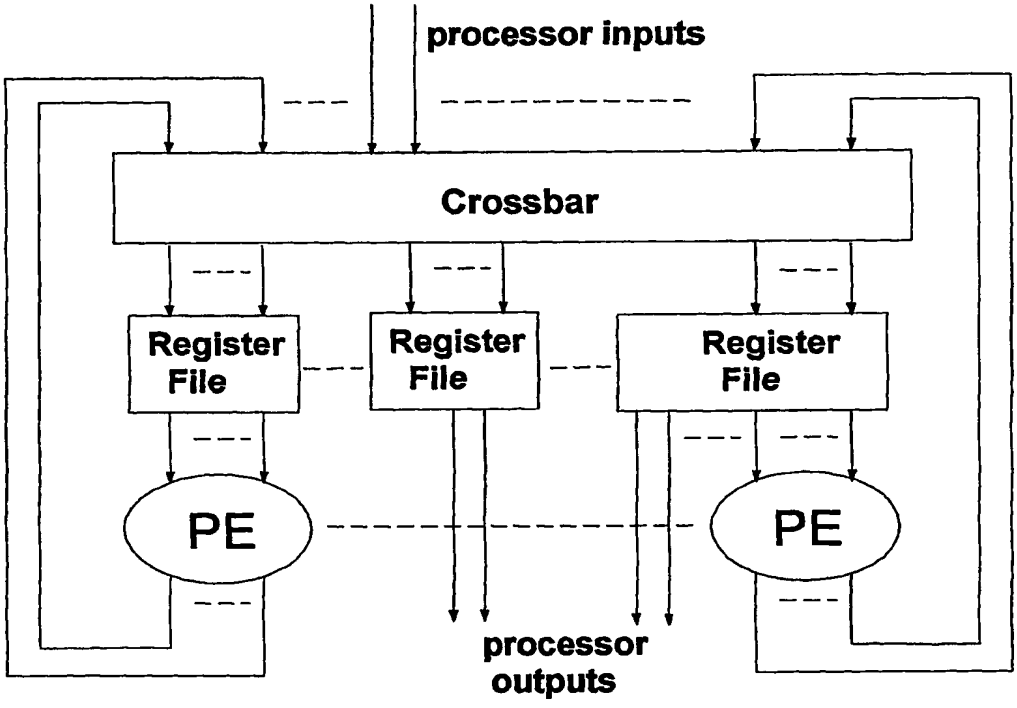


Figure 4

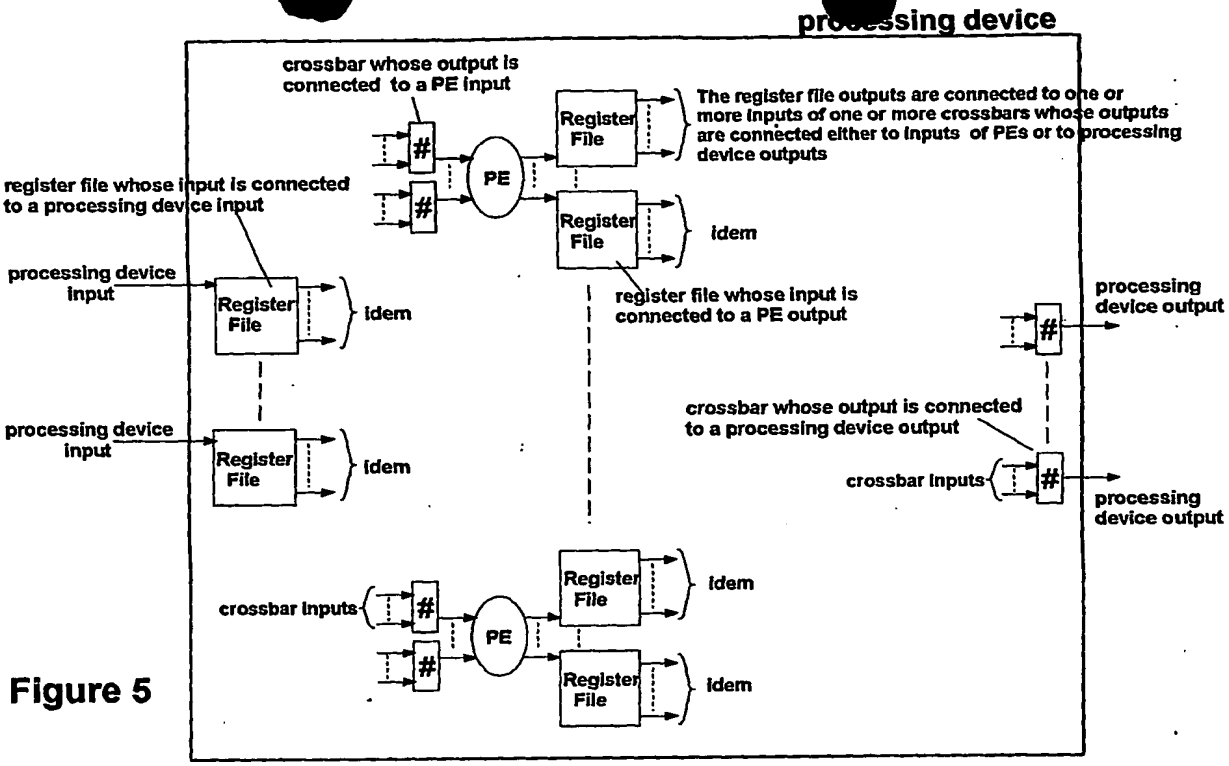


Figure 5

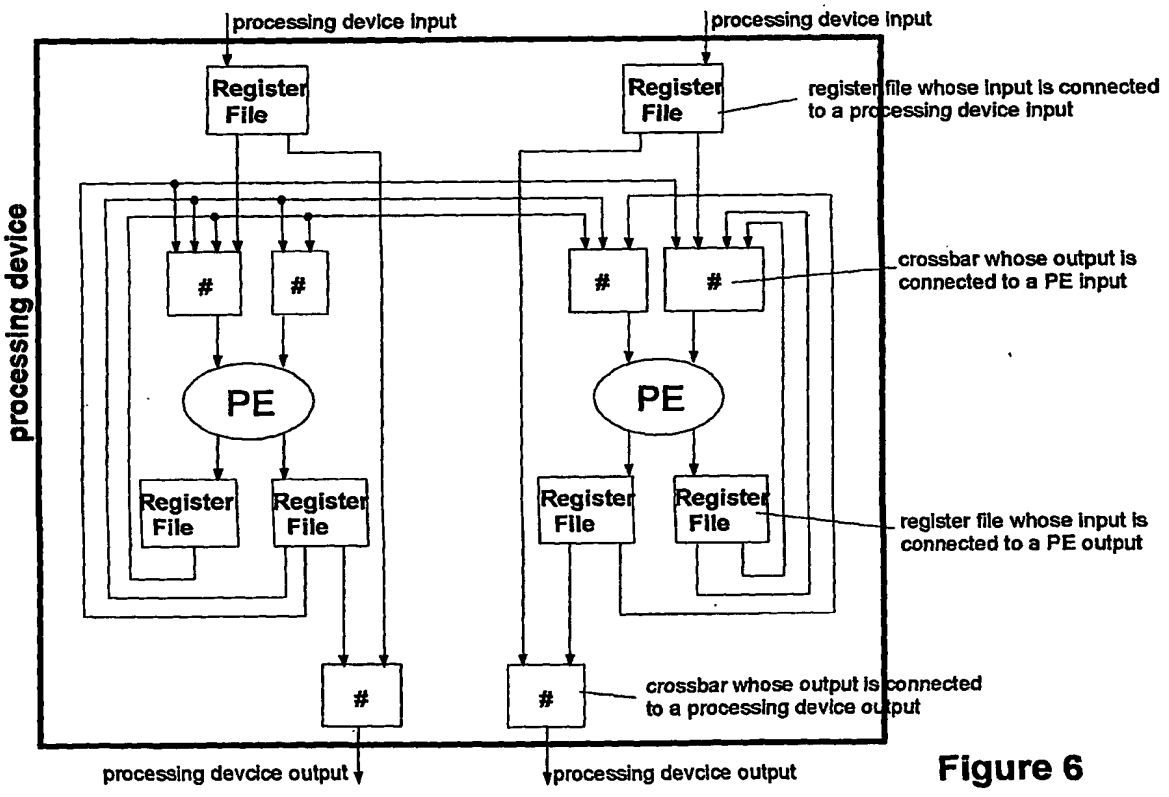


Figure 6

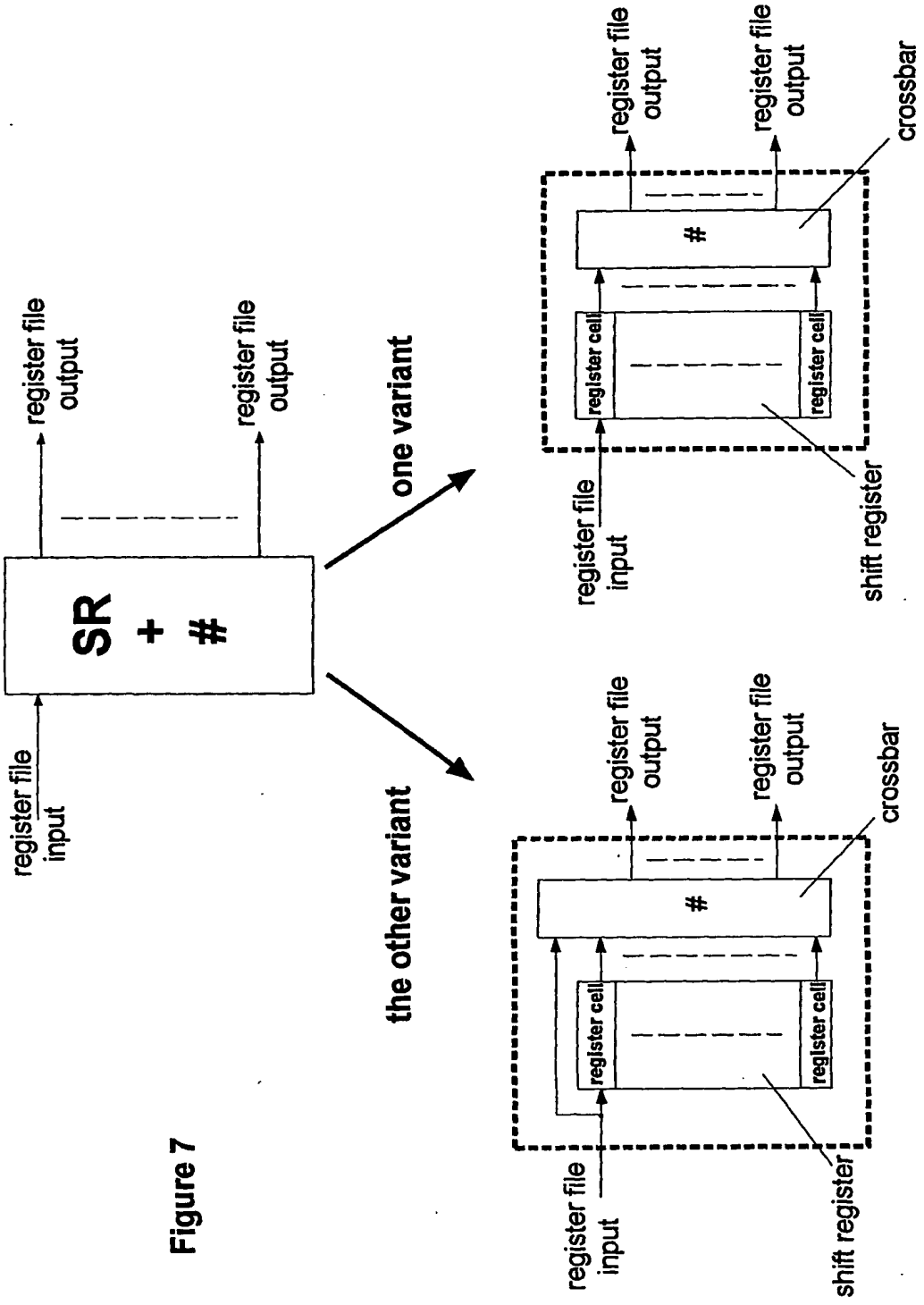

register file output
register file output
SR + #
register file input
one variant
the other variant
register file output
register file output
crossbar
#
register cell
register cell
register file input
shift register
register file output
register file output
crossbar
#
register cell
register cell
register file input
shift register

Figure 7

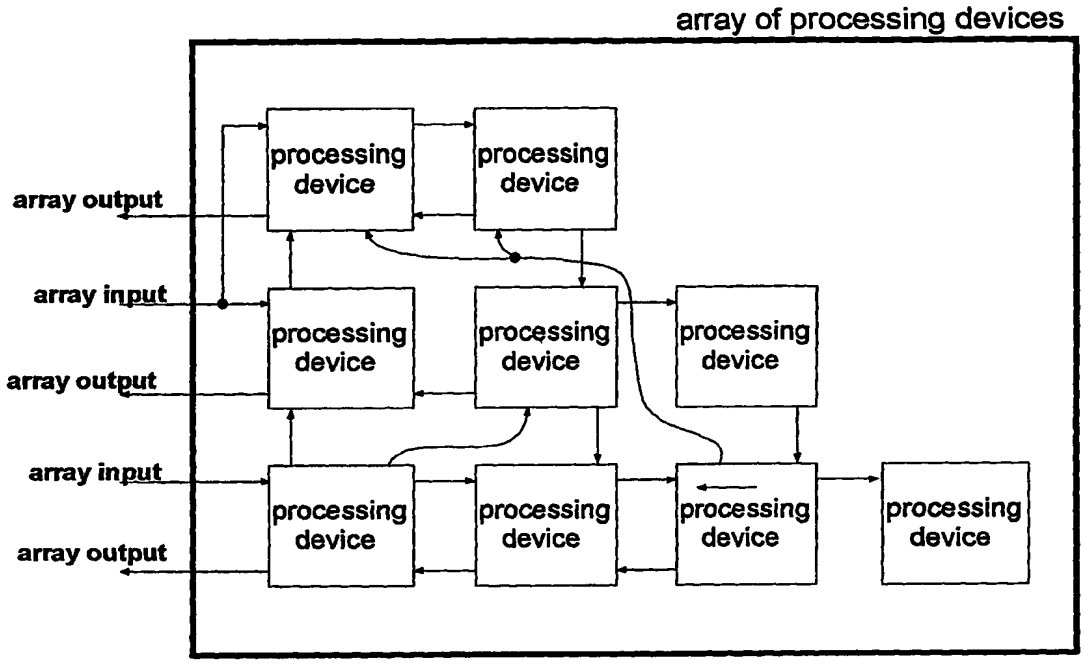

array of processing devices
array output
array input
array output
array input
array output
processing device
processing device
processing device
processing device
processing device
processing device
processing device
processing device
processing device

Figure 8

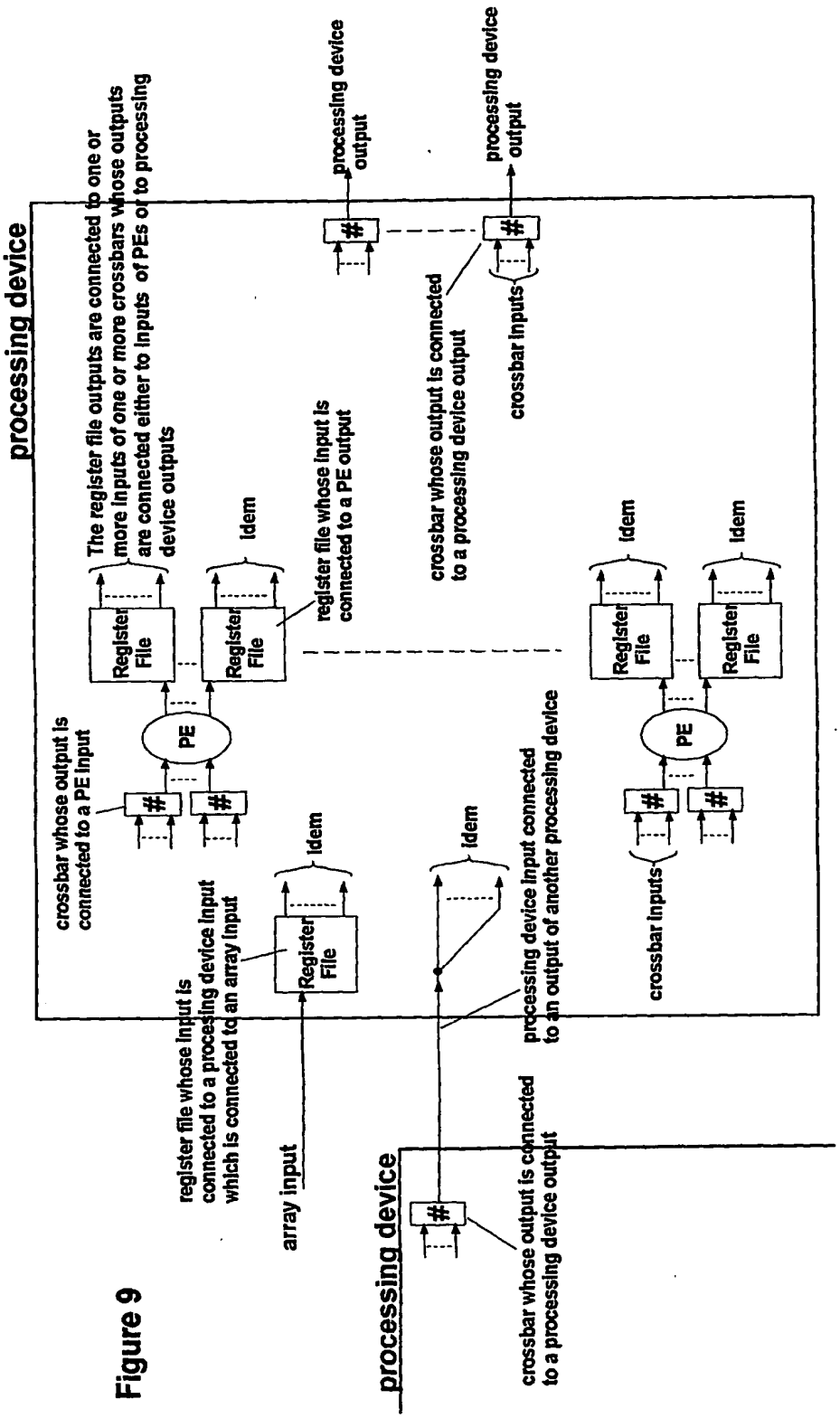

processing device
The register file outputs are connected to one or more inputs of one or more crossbars whose outputs are connected either to inputs of PEs or to processing device outputs
processing device output
processing device output
crossbar whose output is connected to a processing device output
crossbar inputs
register file whose input is connected to a PE output
Idem
Register File
PE
crossbar whose output is connected to a PE input
register file whose input is connected to a processing device input which is connected to an array input
array input
processing device input connected to an output of another processing device
crossbar inputs
crossbar whose output is connected to a processing device output
processing device

Figure 9

INTERNATIONAL SEARCH REPORT

International Application No
PCT/EP 00/00259

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F9/38 G06F9/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ERCANLI E ET AL: "A REGISTER FILE AND SCHEDULING MODEL FOR APPLICATION SPECIFIC PROCESSOR SYNTHESIS" LAS VEGAS, JUNE 3 - 7, 1996,NEW YORK, IEEE,US, vol. CONF. 33, 3 June 1996 (1996-06-03), pages 35-40, XP000640319 ISBN: 0-7803-3294-6 the whole document	1-14
A	US 5 692 139 A (LABROUSSE JEAN-MICHEL J ET AL) 25 November 1997 (1997-11-25) figure 1	1,8
A	EP 0 657 802 A (TEXAS INSTRUMENTS INC) 14 June 1995 (1995-06-14) claim 1; figure 8	2-7,9-14

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 October 2000	02/11/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Daskalakis, T

1

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	MAKOTO IKEDA ET AL: "DATA BYPASSING REGISTER FILE FOR LOW POWER MICROPROCESSOR" IEICE TRANSACTIONS ON ELECTRONICS,JP,INSTITUTE OF ELECTRONICS INFORMATION AND COMM. ENG. TOKYO, vol. E78-C, no. 10, 1 October 1995 (1995-10-01), pages 1470-1472, XP000550056 ISSN: 0916-8524	

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int[redacted] Application No
PCT/EP 00/00259

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5692139 A	25-11-1997	NL 8800053 A	01-08-1989
		DE 68909425 D	04-11-1993
		DE 68909425 T	07-04-1994
		EP 0325310 A	26-07-1989
		ES 2047103 T	16-02-1994
		FI 890066 A,B,	12-07-1989
		HK 20295 A	24-02-1995
		JP 1217575 A	31-08-1989
		US 5103311 A	07-04-1992
		DE 69130723 D	18-02-1999
		DE 69130723 T	22-07-1999
		EP 0479390 A	08-04-1992
		JP 4299436 A	22-10-1992
		US 5832202 A	03-11-1998
		US 5313551 A	17-05-1994
		US 5862399 A	19-01-1999
EP 0657802 A	14-06-1995	US 6067613 A	23-05-2000
		JP 8006544 A	12-01-1996

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.